Deformable-model based Textured Object Segmentation

Xiaolei Huang[1], Zhen Qian[2], Rui Huang[1], Dimitris Metaxas[1,2]

{xiaolei, ruihuang, dnm}@cs.rutgers.edu, zqian@eden.rutgers.edu

[1] Division of Computer and Information Sciences, Rutgers University, NJ, USA

[2] Department of Biomedical Engineering, Rutgers University, NJ, USA

Abstract. In this paper, we present a deformable-model based solution for segmenting objects with complex texture patterns of all scales. The external image forces in traditional deformable models come primarily from edges or gradient information and it becomes problematic when the object surfaces have complex large-scale texture patterns that generate many local edges within a same region. We introduce a new textured object segmentation algorithm that has both the robustness of model-based approaches and the ability to deal with non-uniform textures of both small and large scales. The main contributions include an information-theoretical approach for computing the natural scale of a "texon" based on model-interior texture, a nonparametric texture statistics comparison technique and the determination of object belongingness through belief propagation. Another important property of the proposed algorithm is in that the texture statistics of an object of interest are learned online from evolving model interiors, requiring no other *a priori* information. We demonstrate the potential of this model-based framework for texture learning and segmentation using both natural and medical images with various textures of all scales and patterns.

1. Introduction

Deformable models, which are curves or surfaces that move under the influence of internal smoothness forces and external image forces, have been extensively studied and widely used for robust object segmentation. In the literature, there are two major classes of deformable models. One is the parametric deformable models [12, 25, 5] that explicitly represent curves and surfaces in their parametric form; the other class is the geometric deformable models [3, 14], which implicitly represent model shape as the zero level set of a higher-dimensional scalar function, and evolve the model based on front propagation using the theory of curve evolution. In the formulations for both types of models, the external image forces traditionally come from edge or image gradient information. However this makes the models unfit for finding boundaries of objects with complex large-scale texture patterns due to the presence of large variations in image gradient and many local edges inside the object.

There have been efforts to address this problem. Geodesic Active Regions [17] deal with supervised texture segmentation in a frame partition framework using level-set deformable model implementation. There are several assumptions in this supervised method however, which include knowing beforehand the number of regions in an image and the statistics for each region are learned off-line using a mixture-of-gaussian approximation. These assumptions limit the applicability of the method to a large variety of natural images. Region Competition [27] performs texture segmentation by combining region growing and active contours using multi-band input after applying a set of texture filters. The method assumes multivariate Gaussian distributions on the filter-response vector inputs. Another texture segmentation approach in a deformable model framework [20] is based on deforming an improved active contour model [26] on a likelihood map instead of heuristically constructed edge map. However, because of the artificial neighborhood operations, the results from this approach suffer from blurring on the likelihood map, which causes the object boundary detected to be "dilated" versions of the true boundary. The dilate zone could be small or large depending on their neighborhood size parameter. Metamorphs [9] is recently proposed as a new class of deformable models that integrate boundary information and nonparametric region statistics for segmentation. It mostly dealt with intensity images with noise and complex intensity distributions however.

In this paper, we propose a new deformable-model based solution to accurate and robust texture segmentation. The key novelty and contribution are in the analysis and representation of model-interior texture statistics. The first step in analysis is to determine a best natural scale for the texons, which are the basic structural elements for the texture of interest. If this scale is small, then a nonparametric kernel-based approximation of the model-interior intensity distribution is sufficient to capture the statistics; if the scale is large which often means the texture consists of large-scale periodic patterns, a small bank of Gabor filters is applied and the nonparametric statistics are evaluated on the filter responses. We then compute the likelihood map of texture consistency over the entire image domain by comparing local patch statistics with the model-interior statistics. This type of analysis enables our method to automatically and adaptively deal with both regions that are mostly homogeneous (intensity) and regions that consist of large-scale patterns, while keeping a sharp edge on the likelihood map right on the texture region boundary. The deformable model dynamics is derived from energy terms defined on the computed likelihood map, and when the model evolves, the model-interior statistics are re-analyzed and the likelihood map is updated. This adaptive online-learning process on one hand constraints the model to deformations that keep consistent model-interior statistics, and on the other hand enables the model to converge to true object boundary in the presence of not-perfectly uniform texture patterns.

1.1. Previous Work

Texture analysis and segmentation is an important problem in many areas of computer vision, because images of natural scenes, medical images and images of many other modalities are mainly composed of textured objects. Different texture segmentation approaches have been presented in the literature and they typically follow two steps: a modelling step to generate texture descriptors to describe texture appearance, and an optimization step to group pixels into regions of homogeneous texture appearance. In the modelling phase, Markov Random Fields [15], banks of filters [21], wavelets [2], etc. are some common techniques. In the classification/grouping phase, supervised methods [17, 2] partition an image by maximizing likelihood given features or statistics learned from a set of texture patterns given *a priori*, while un-supervised methods [15, 13] apply clustering techniques to group pixels into homogeneous segments in the descriptor vector space.

Recently Graph Cuts [24] has been coupled with contour and texture analysis [13] to achieve un-supervised segmentation on varieties of images, and one of its extension, GrabCut [22], has shown promising results in interactive textured foreground extraction. Another work is Active Shape and Appearance Models [7, 6], which learns statistical models of object shape and appearance and use the prior models to segment textured objects.

Compared to previous work, the deformable model based texture segmentation method we propose integrates high-level model constraints with low-level processing and classification. It has the advantage of not requiring off-line supervised learning, yet it enjoys the benefits of having a likelihood map measuring texture consistency by means of online adaptive learning of model-interior statistics. Another advantage is in its elaborate information-theoretic texon scale analysis, which eliminates the classic blurring effect around texture boundaries on computed likelihood maps.

The remainder of the paper is organized as follows. We present an overview of our method in section 2. Then detailed algorithm in each step is described in section 3. Experimental results are presented in section 4, and we conclude with discussions in section 5.

2. Overview

The basic idea of our deformable-model based texture segmentation algorithm is depicted in Fig. 1, and the algorithm consists of the following steps:

1. Initialize a simple-shape deformable model centered around a seed point. In Fig. 1(a), the initial model is the larger circular model, and the seed point is marked by an asterisk.

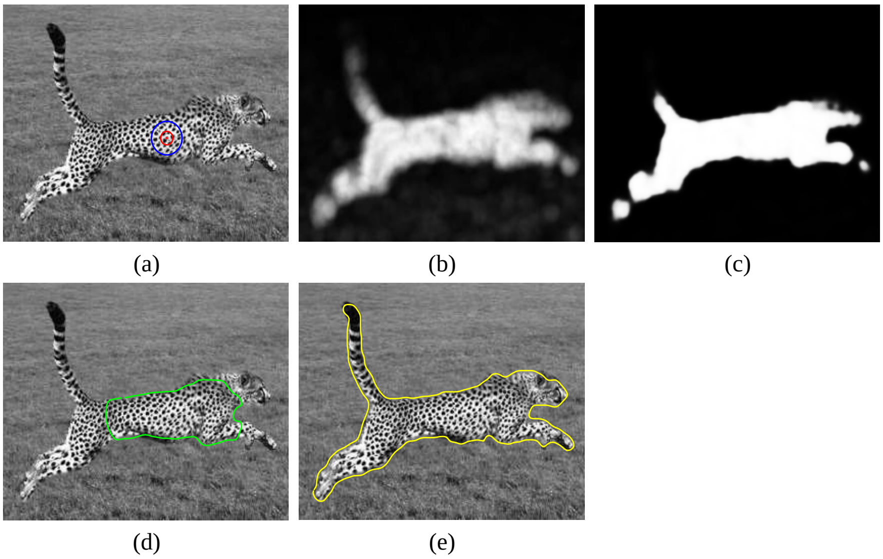

Fig. 1. (a) the original cheetah image. Initial Model: outer large circle; Texon scale: inner small circle, (b) likelihood map computed based on texon statistics, (c) updated likelihood map after applying BP based MRF, (d) an intermediate model, (e) converged model.

2. Determine a “best” natural scale for the texture elements that are basic building blocks of the model interior texture. We call such texture elements “texons”, following the naming convention in [10, 13, 28]. In Fig. 1(a), the determined texon scale is indicated by the inner smaller circle.
3. Compute nonparametric statistics of the model-interior texons, either on intensity or on Gabor filter responses depending on the texon scale, and compare it with local-neighborhood statistics surrounding each pixel in the image. Thereafter a likelihood map is generated which measures the consistency of each local patch texture with the model-interior texture. Fig. 1(b) shows an example of such computed likelihood map given the model in Fig. 1(a).
4. Use the Belief-Propagation implementation of Markov Random Fields to update the likelihood map, taking into account contextual information from neighboring texture patches. For the cheetah image, the updated likelihood map is shown in Fig. 1(c).
5. Evolve the deformable model, for a fixed number of iterations, toward object boundary based on model dynamics derived from both texture energy terms defined on the likelihood map and balloon-force energy terms defined on the model geometry. In Fig. 1(d), an intermediate model is drawn on the image.
6. Repeat steps 3-5 until convergence (e.g. when model deformation between iterations is sufficiently small). The converged model on the cheetah image is shown in Fig. 1(e).

3. Methodology

In this section, we present the detailed algorithm in each step.

3.1. Implicit Model Representation

In our framework, we use the Euclidean distance transform to embed an evolving deformable model as the zero level set of a higher dimensional distance function [16, 18, 9]. In the 2D case, let $\Phi : \Omega \rightarrow R^{+}$ be a Lipschitz function that refers to the distance transform for the model shape $\mathcal{M}$. The shape defines a partition of the domain: the region that is enclosed by $\mathcal{M}$, $[\mathcal{R}_{\mathcal{M}}]$, the background $[\Omega - \mathcal{R}_{\mathcal{M}}]$, and on the model, $[\partial \mathcal{R}_{\mathcal{M}}]$. Given these definitions the following implicit shape representation is considered:

$$\Phi_{\mathcal{M}}(\mathbf{x}) = \begin{cases} 0, & \mathbf{x} \in \partial \mathcal{R}_{\mathcal{M}} \\ +E_d(\mathbf{x}, \mathcal{M}) > 0, & \mathbf{x} \in \mathcal{R}_{\mathcal{M}} \\ -E_d(\mathbf{x}, \mathcal{M}) < 0, & \mathbf{x} \in [\Omega - \mathcal{R}_{\mathcal{M}}] \end{cases} \quad (1)$$

where $E_d(\mathbf{x}, \mathcal{M})$ refers to the min Euclidean distance between the image pixel location $\mathbf{x} = (x, y)$ and the model $\mathcal{M}$.

This implicit representation makes the model shape representation a distance map “image”, which greatly facilitates the integration of boundary and region information. It also offers topology freedom and parameterization independence during model evolutions.

3.2. Texture Consistency Likelihood map given Model Interior

Given the current model shape, our first goal is to find a mathematical solution to determine a “best” local scale for the texture elements that are basic building blocks of the model interior texture.

Best Local Scale for Model Interior Texons We approach this scale problem using a detector based on comparing a texon interior intensity probability density function (p.d.f.) with the whole model interior p.d.f., and we determine the scale of the texon as the smallest scale that provides a texon p.d.f that is sufficiently close to the overall model interior p.d.f.

Suppose the model is placed on image I, and the image region bounded by current model $\Phi_{\mathcal{M}}$ is $\mathcal{R}_{\mathcal{M}}$, we use a nonparametric kernel-based method to approximate the p.d.f. of the model interior intensity. Let us denote the random variable for intensity values by $i, i = 0, ..., 255$, then the intensity p.d.f. of the model-interior region is defined by:

$$\mathbf{P}(i \big| \Phi_{\mathcal{M}}) = \frac{1}{V(\mathcal{R}_{\mathcal{M}})} \iint_{\mathcal{R}_{\mathcal{M}}} \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-(i - I(\mathbf{y}))^2}{2\sigma^2}} d\mathbf{y} \quad (2)$$

where $V(\mathcal{R}_{\mathcal{M}})$ denotes the volume of $\mathcal{R}_{\mathcal{M}}$, $\mathbf{y}$ are pixels in the domain $\mathcal{R}_{\mathcal{M}}$, and σ is a constant specifying the width of a gaussian kernel.

Similarly, the intensity p.d.f. for a local texon can be defined as in Eq. 2, the only difference being that the integration is over pixels inside the texon.

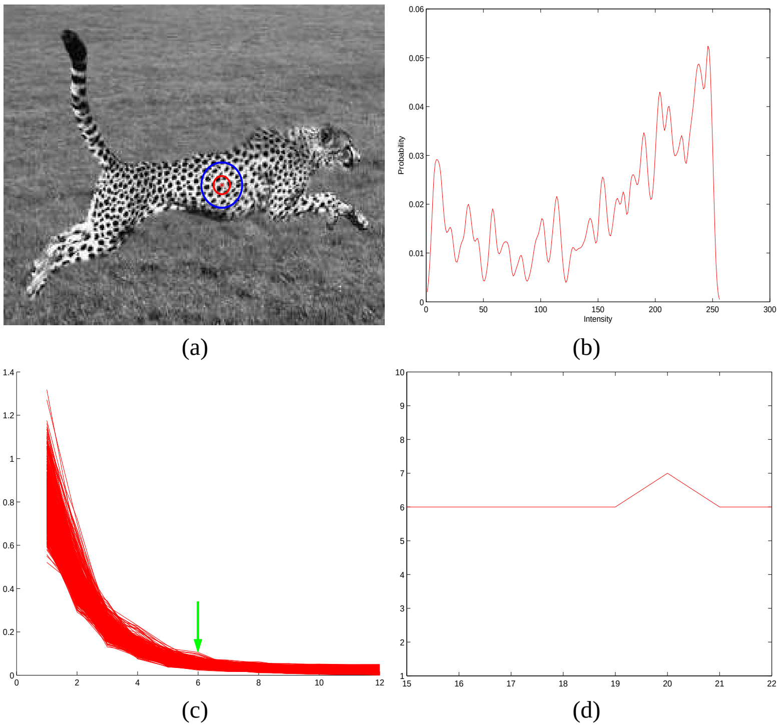

Fig. 2. (a) Cheeta image. Outer large circle shows an initial model, inner small circle shows the determined best scale for model-interior texons. (b) Overall model interior p.d.f. (c) Y axis: K-L distance between texon p.d.f. and overall model-interior p.d.f.; X axis: changing scale (i.e. radius) of the texon under evaluation. Each curve represents a texon centered at a different pixel inside the model. (d) The best scale determined remains stable as we change the size of the initial model.

To measure the dissimilarity between two probability density functions, we adopt an information-theoretic distance measure, the Kullback-Leibler (K-L) Divergence [1]. Since the K-L divergence is asymmetric, we instead use one of its symmetrized relative – the Chernoff Information. The Chernoff Information between p_1 and p_2 is defined by:

$$C(p_2 \| p_1) = \max_{0 \leq t \leq 1} \ -\log \mu(t)$$

where $\mu(t) = \int [p_1(i)]^{1-t} [p_2(i)]^t di$. A special case of Chernoff ”distance” is the Bhattachayya ”distance”, in which t is chosen to be $\frac{1}{2}$, i.e., the Bhattachayya ”distance” between p_1 and p_2 is:

$$B(p_2 \| p_1) = -\log \mu(\frac{1}{2})$$

In order to facilitate notation, we write:

$$\rho(p_2 \| p_1) = \mu(\frac{1}{2}) = \int [p_1(i)]^{\frac{1}{2}} [p_2(i)]^{\frac{1}{2}} di \qquad (3)$$

Clearly, when the value for ρ ranges from one to zero, the value for B goes from zero to infinity.

The Chernoff Information is an important information-theoretic distance measure, and it has been shown this measure is the exponential rates of optimal classifier performance probabilities [4]. The use of this measure is also justified in our experiments in which we observe a stable convergence in the distances between texon p.d.f and overall model interior p.d.f as the test scale increases.

The steps in determining the scale of texons inside the current model are as follows.

1. Approximate the intensity p.d.f. of the overall model interior (Eq. 2). Denote this p.d.f. as p_m.
   The p.d.f. for the cheetah example based on the initial model in Fig. 2(a) is displayed in Fig. 2(b).
2. Choose a best scale $\hat{s}$ for the model interior texons among all possible scales between $1...S$ [1].
   Let us denote a texon of scale s centered at a pixel $\mathbf{x}$ by $T(\mathbf{x}, s)$, and its interior intensity p.d.f. by $p_{T(\mathbf{x},s)}$. To determine the best scale $\hat{s}$, we compute the Bhattachayya distance between p_m and $p_{T(\mathbf{x},s)}$, for all pixels $\mathbf{x}$ inside the model and for all scales $s = 1...S$. Fig. 2(c) visualizes the functional relationship between such distances and the scale in a graph. In the graph, each curve represents the "distance-scale" function for texons centered at a different pixel. From the graph, we can see that, as the scale increases, the Bhattachayya distance decreases asymptotically at all pixels, and all curves finally converge at a small value. This behavior proves the validity of the usage of this symmetrized K-L distance measure, and it also exposes to us a way to determine the natural scale of the model interior texons – the scale corresponding to the point of inflection on the Distance-Scale function curves. Since we get a scale value for every pixel inside the model this way, we use a robust estimator, the median estimator, to choose the best scale $\hat{s}$ as the median of the inflection-point scales chosen for all these pixels. On Fig. 2(a), we indicate the best scale computed this way by the inner small circle.

Based on our experiments, this ”best” natural scale for model-interior texons determined using the method above is invariant to the size of the initial model. Fig. 2(d) shows the functional relation between the best scale chosen vs. initial size of the model for the cheetah example. We can see from the curve that the best scale remains stable as the size of the initial model changes. This behavior is also observed in many other examples that we tested.

Compute Texture Likelihood map Once we have determined the scale for model-interior texons, we can, for every pixel on the image, evaluate the likelihood of its neighborhood texon being consistent with the object texture learned from the model interior texture. We define this likelihood using the ρ value in Eq. 3, since it increases as the Bhattachayya distance between two distributions decreases. That is, the likelihood of any pixel $\mathbf{x}$ on the image I belonging to the object of interest is defined by:

$$\mathcal{L}(T(\mathbf{x}, \hat{s}) \big| \varPhi_{\mathcal{M}}) \propto \rho(p_{T(x,\hat{s})} || p_m) \tag{4}$$

[1] Here we assume that the current model interior contains at least one texon, and the largest test scale S is smaller than the size of the model.

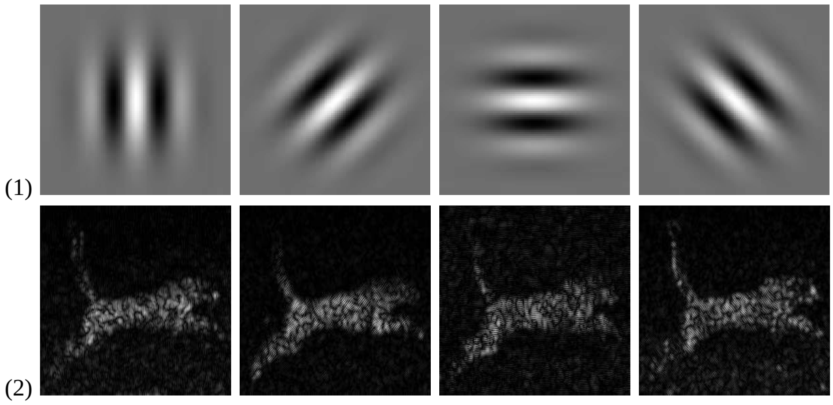

Fig. 3. (1) Gabor filters in a small bank with constant frequency and shape, and varying orientation. (2) Responses of the cheetah image to Gabor filters in (1).

where $T(\mathbf{x}, \hat{s})$ represents the neighborhood texon centered at $\mathbf{x}$ and with scale $\hat{s}$, $p_{T(x,\hat{s})}$ is the intensity p.d.f. of the texon intensity, and p_m is the p.d.f. learned from interior intensity of the model initialized inside the object.

One limitation of using the nonparametric intensity p.d.f. to approximate texon(or model) interior texture statistics is that, the information on pixel order and spatial correlation between pixels within a texon is lost. For instance, if we take a texon inside the object, randomly re-permute all pixels within it to generate a new texon, then copy this new texon to locations surrounding the object, then the computation in Eq. 4 would have trouble differentiating these two kinds of texons, even though they appear different.

The importance of the texture pattern (i.e. pixel order) information depends on the scale $\hat{s}$ though, since this scale reveals to some extent the characteristics of the model-interior texture. In our framework, we separate two cases according to $\hat{s}$ and treat them differently when computing the likelihood map.

C1: In the first case, if $\hat{s}$ is very small (e.g. the radius of texons is less than 3 pixels wide), the model-interior texture is mostly homogeneous with some level of noise, hence it is not necessary to further consider the spatial correlation between pixels, and Eq. 4 should be sufficient in this case.

C2: In the second case, if $\hat{s}$ is rather large, we predict that the model-interior texture consists of periodic mosaics of large-scale patterns, and additional measures are necessary to capture the statistics in these patterns as well. We approach this by applying a small number of Gabor filters [8] to the model interior, and learn statistics with respect to the Gabor responses.

In our experiments, we choose $N(N = 4)$ Gabor filter bases (Fig. 3(1)). The filters have constant frequency and shape, but with varying orientations. The frequency and Gaussian-envelop shape of the filters are computed based on the predetermined scale of model-interior texons. For each of the N Gabor filters in the small filter bank, we get a response image $R_n, n = 1, ..., N$, as shown in Fig. 3(2). Suppose the random variable for the response value from filter n is X_n, we learn the probability density function of X_n, using the same nonparametric kernel-based ap-

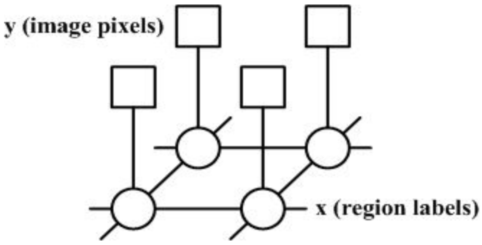

Fig. 4. The MRF Graphical Model.

proximation as in Eq. 2, for pixels inside the model. This way, we have N p.d.f.s, $p_n^{\mathcal{M}}, n = 1, ..., N$, to describe the statistics of the model-interior response to the different Gabor filters.

When computing the likelihood map for pixels of the entire image, we use a mechanism similar to that in Eq. 4, but we combine the probabilities derived from all Gabor filter responses. That is, for every pixel $\mathbf{x}$ on the image, we take its local neighborhood texon $T(\mathbf{x}, \hat{s})$, and compute its interior response-value p.d.f.s $p_n^T, n = 1, ..., N$. Then the likelihood of this texon being consistent with the model-interior texture pattern is measured by:

$$\begin{aligned} \mathcal{L}(T(\mathbf{x}, \hat{s}) \big| \Phi_{\mathcal{M}}) &\propto P(T \big| p_1^{\mathcal{M}}, ..., p_N^{\mathcal{M}}) \\ &= \textstyle\prod_{n=1}^{N} P(T \big| p_n^{\mathcal{M}}) \\ &\propto \textstyle\prod_{n=1}^{N} \rho(p_n^T \| p_n^{\mathcal{M}}) \end{aligned} \tag{5}$$

In the equation, we approximate the likelihood in terms of each Gabor filter response using the ρ value, and the relations can be easily derived given that the responses from different Gabor-filter bases are conditionally independent of each other.

After computing the likelihood at each pixel over the entire image domain, we denote the resulting likelihood map $\mathcal{L}(T(\mathbf{x}, \hat{s}) \big| \Phi_{\mathcal{M}})$ as L_I.

Contextual Confirmation through belief propagation The likelihood map L_I quantifies the probability of every local texon belonging to part of the texture region of interest. However, all measurements are still local, and no context information between neighboring texons is accounted for. Markov Random Field (MRF) models are often used to capture dependencies between neighboring cliques (e.g. pixels, texons, etc.), and can be applied on the likelihood map to reduce noise and improve neighborhood consistency.

Given a typical graphical-model illustration for MRF, as shown in Fig. 4, the graph has two kinds of nodes: hidden nodes (circles in Fig. 4, representing region labels) and observable nodes (squares in Fig. 4, representing image pixels). Edges in the graph depict relationships between the nodes.

Let n be the number of the hidden/observable states (i.e., the number of pixels in the image). A configuration of the hidden layer is:

$$\mathbf{h} = (h_1, ..., h_n), h_i \in V, i = 1, ..., n \tag{6}$$

where V is a set of region labels, such as $V = 0, 1$, where the value 0 indicates different texture from the model interior, and the value 1 indicates same texture as the model interior.

Similarly, a configuration of the observable layer is:

$$\mathbf{o} = (o_1, ..., o_n), o_i \in D, i = 1, ..., n \tag{7}$$

where D is a set of pixel values, e.g., the original likelihood values in the map L_I. The relationship between the hidden states and the observable states (also known as local evidence) can be represented as the compatibility function:

$$\phi(h_i, o_i) = P(o_i|h_i) \tag{8}$$

Similarly, the relationship between the neighboring hidden states can be represented as the second compatibility function:

$$\psi(h_i, h_j) = P(h_i, h_j) \tag{9}$$

Now the inference problem can be viewed as a problem of estimating the MAP solution of the MRF model:

$$\mathbf{h}_{MAP} = argmax_{\mathbf{h}} P(\mathbf{h}|\mathbf{o}) \tag{10}$$

where

$$P(\mathbf{h}|\mathbf{o}) \propto P(\mathbf{o}|\mathbf{h})P(\mathbf{h}) \propto \prod_i \phi(h_i, o_i) \prod_{(i,j)} \psi(x_i, x_j) \tag{11}$$

The exact MAP inference in MRF models is computationally infeasible, and we use an approximation technique based on the Belief Propagation (BP) algorithm, which is an inference method proposed by [19] to efficiently estimate Bayesian beliefs in the network by iteratively passing messages between neighbors. We assume the likelihood values in each region follow a Gaussian distribution:

$$\phi(h_i, o_i) = \frac{1}{\sqrt{2\pi\sigma_{x_i}^2}} exp\big(-\frac{(o_i - \mu_{x_i})^2}{2\sigma_{x_i}^2}\big) \tag{12}$$

and the compatibility function between neighboring hidden states is represented by:

$$\psi(o_i, o_j) = \frac{1}{Z} exp\big(\frac{\delta(o_i - o_j)}{\sigma^2}\big) \tag{13}$$

where $\delta(x) = 1$ if $x = 0$; $\delta(x) = 0$ if $x \neq 0$, σ controls the degree of similarity between neighboring hidden states, and Z is a normalization constant.

After this step of MRF contextual confirmation, the resulting new likelihood map is denoted by L_I^c. One example demonstrating the effect of this step can be seen in Fig. 1(c). In our experiments, we use the $\{0, 1\}$ region labels as the hidden states, hence by thresholding at 0.5, we can differentiate regions that have similar texture with the model-interior from other background regions.

3.3. Deformable Model Dynamics

In order to evolve the deformable model toward the boundary of the texture region of interest, we derive the model dynamics in a variational framework by defining energy terms leading to both external texture/image forces and internal balloon forces.

Free Form Deformations The deformations a model in our framework can undergo are defined using a space warping technique, the Free Form Deformations (FFD) [23, 9]. The essence of FFD is to deform an object by manipulating a regular control lattice F overlaid on its volumetric embedding space, hence it integrates naturally with the implicit model shape representation (see section 1). In the Incremental FFD formulation used in [9], the deformation parameters, $\mathbf{q}$, are the deformations of the control points in both x and y directions:

$$\mathbf{q} = \{(\delta F^x_{m,n}, \delta F^y_{m,n})\};\ (m,n) \in [1,M] \times [1,N]$$

where the control lattice is of size $M \times N$. The deformed position of a pixel $\mathbf{x} = (x,y)$, given the deformation of the control lattice from an initial regular configuration F^0 to a new configuration F, is defined in terms of a tensor product of Cubic B-spline polynomials:

$$D(\mathbf{q};\mathbf{x}) = \mathbf{x} + \delta D(\mathbf{q};\mathbf{x}) = \sum_{k=0}^{3}\sum_{l=0}^{3} B_k(u)B_l(v)(F^0_{i+k,j+l} + \delta F_{i+k,j+l}) \tag{14}$$

where $i = \lfloor \frac{x}{X} \cdot (M-1) \rfloor + 1,\ j = \lfloor \frac{y}{Y} \cdot (N-1) \rfloor + 1.$

Since FFD imposes implicit smoothness constraints during model deformation, which guarantees C^1 continuity at control points and C^2 continuity everywhere else, we omit model smoothness energy terms that are common in traditional parametric or level-set based deformable models.

Data Terms for Texture/Image forces Given the likelihood map L^c_I computed based on the current model-interior texture statistics, we are able to segment out foreground regions that have similar texture with the model-interior (see section 3.2) by thresholding at 0.5 on L^c_I. Since there may be many disconnected foreground regions detected this way, we choose only the one overlapping the current model as the current region of interest (ROI). Suppose the binary mask of this ROI is I_r, we encode its boundary information by computing the Euclidean distance transform of I_r, which is denoted by Φ_r. Then we define a data energy term to evolve the model toward the ROI boundary as follows:

$$E_{data} = \frac{1}{V(\mathcal{R}_\mathcal{M})} \iint_{\mathcal{R}_\mathcal{M}} \big(\Phi_\mathcal{M}(\mathbf{x}) - \Phi_r(D(\mathbf{q};\mathbf{x}))\big)^2 d\mathbf{x}$$

where $\Phi_\mathcal{M}$ is the implicit representation of the current model (Eq. 1), $\mathcal{R}_\mathcal{M}$ is the model interior region, $V(\mathcal{R}_\mathcal{M})$ refers to the volume of region $\mathcal{R}_\mathcal{M}$, and $D(\mathbf{q};\mathbf{x})$ is the FFD definition for the position of a sample pixel $\mathbf{x}$ after deformation (Eq. 14).

Energy Term for Balloon forces One additional energy term is defined on the model geometry to explicitly grow the model along its normal direction, which can expedite the model convergence process. It is also very important for accurate model convergence when the shape of the texture region has salient protrusions or concavities. The definition for the balloon-force energy term is as follows:

$$E_{balloon} = \frac{1}{V(\partial\mathcal{R}_{\mathcal{M}})} \iint_{\partial\mathcal{R}_{\mathcal{M}}} \big(\varPhi_{\mathcal{M}}(D(\mathbf{q};\mathbf{x}))\big) d\mathbf{x}$$

where $\partial\mathcal{R}_{\mathcal{M}}$ refers to the model affinity, which in practice we take as a narrow band around the model $\mathcal{M}$ (i.e. zero level set of the model representation $\varPhi_{\mathcal{M}}$). The reason behind the form of this term is because of the definition for $\varPhi_{\mathcal{M}}$, which has negative values outside the model, zero value on the model, and positive values inside the model.

Model Evolution In our formulations above, both the data term and the balloon term are differentiable with respect to the model deformation parameters $\mathbf{q}$, hence a unified gradient-descent based parameter updating scheme can be derived. Let the overall energy functional be:

$$E = E_{data} + kE_{balloon} \tag{15}$$

where k is a constant balancing the contributions of the two terms. Then the following evolution equation for each element $\mathbf{q}_i$ in the model deformation parameters $\mathbf{q}$ can be derived:

$$\frac{\partial E}{\partial \mathbf{q}_i} = \frac{\partial E_{data}}{\partial \mathbf{q}_i} + k\frac{\partial E_{balloon}}{\partial \mathbf{q}_i} \tag{16}$$

where

$$\begin{aligned}\frac{\partial E_{data}}{\partial \mathbf{q}_i} = \frac{1}{V(\mathcal{R}_{\mathcal{M}})} \iint_{\mathcal{R}_{\mathcal{M}}} & 2\ \varPhi_{\mathcal{M}}(\mathbf{x}) - \varPhi_r(D(\mathbf{q};\mathbf{x}))\ \cdot \\ & - \nabla\varPhi_r(D(\mathbf{q};\mathbf{x})) \cdot \frac{\partial}{\partial \mathbf{q}_i} D(\mathbf{q};\mathbf{x})\ d\mathbf{x}\end{aligned}$$

$$\frac{\partial E_{balloon}}{\partial \mathbf{q}_i} = \frac{1}{V(\partial\mathcal{R}_{\mathcal{M}})} \iint_{\partial\mathcal{R}_{\mathcal{M}}} \nabla\varPhi_{\mathcal{M}}(D(\mathbf{q};\mathbf{x})) \cdot \frac{\partial}{\partial \mathbf{q}_i} D(\mathbf{q};\mathbf{x})\ d\mathbf{x}$$

In the above formulas, the partial derivatives with respect to the deformation (FFD) parameters, $\frac{\partial}{\partial \mathbf{q}_i} D(\mathbf{q};\mathbf{x})$, can be easily derived from the model deformation formula Eq. 14.

One important advantage of our framework is that, as the model evolves, the model interior changes, hence the model-interior texture statistics get updated and the new statistics are used for further model evolution. This online learning property enables our deformable model framework to segment objects with non-uniform texture patterns to some extent. In Fig. 5, we show the evolution in the likelihood map as the model evolves from an initial circular model to an intermediate model.

Change of Topology: Merging of Multiple Models When multiple models are initialized in an image, each model evolves based on its own dynamics. During evolution, the models may overlap with each ohter, so a collision detection step is needed to check

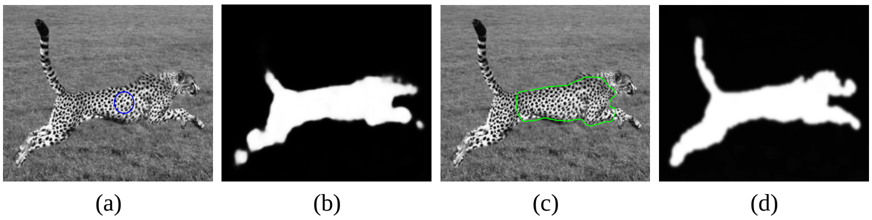

Fig. 5. (a) Initial model. (b) Likelihood map (after MRF) based on initial model. (c) An intermediate model. (d) Likelihood map re-computed based on the intermediate model.

whether the interiors of more than one models overlap. If a collision is detected, the models involved are tested based on their interior texture statistics, and they are merged only if their statistics are sufficiently close.

Suppose a collision is detected between model K and model L. We first compare their model-interior texon scales. If the scales are very different, we do not merge the two models. If the scales are close, we further test their texture statistics. Let us denote their intensity p.d.f.s on the original image I by p_k and p_l respectively. Then we measure the Bhattachayya distance $B(p_k \| p_l)$ between the two distributions. Here we need a definite threshold to determine whether the two distributions are sufficiently close enough. As discussed in [11], the error probability [2], P_e, of two distributions or processes, is related to the $\rho(p_k \| p_l)$ value by the following formula: $\frac{1}{8}\rho^2 \leq P_e \leq \frac{1}{2}$. In our algorithm, by allowing a 10% error probability (i.e., $P_e = 0.1$), we can derive the following threshold on the ρ value: $T_\rho \geq \sqrt{(8 * 0.1)} \approx 0.9$. That is, if the ρ value is less than T_ρ, then the two distributions are the result of two different signals (textures) with a 10% error probability; conversely, if the ρ value is greater than T_ρ, then we consider the two distributions are sufficiently close. More strict tests can be done by comparing Gabor filter response statistics.

If we decide the statistics of two models in collision are sufficiently close, we merge the two models. Unlike parametric curve models that may require sophisticated collision detection mechanism, the collision detection in our method is simple since we use the implicit level set representation for model shapes (see Eq. 1). Given the implicit representations of the two old models $\Phi_{\mathcal{M}_1}(\mathbf{x})$ and $\Phi_{\mathcal{M}_2}(\mathbf{x})$, then the implicit representation of the merged new model can be computed directly by:

$$\Phi_{\mathcal{M}}(\mathbf{x}) = \max\left(\Phi_{\mathcal{M}_1}(\mathbf{x}), \Phi_{\mathcal{M}_2}(\mathbf{x})\right)$$

Thereafter this new model evolves in place of the two old models.

Fig. 6 shows an example where we initialize two models, they first evolve on their own then they merge into one new model upon collision.

4. Experimental Results

We have run our algorithm on a variety of images with textures of different patterns and scales. Figures 7-9 show typical segmentation results. In all the cases, we initialize sev-

[2] This is the probability of mistakenly classifying two processes/distributions as the same when they are in fact different.

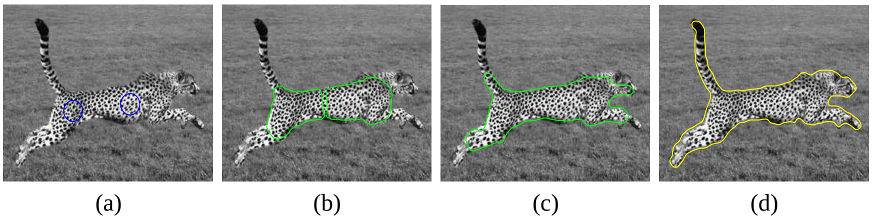

Fig. 6. (a) Two initial models. (b) Two models evolving on their own before merging. (c) The two models are merged into one new model upon collision and the new model continues evolving. (d) The final converged model.

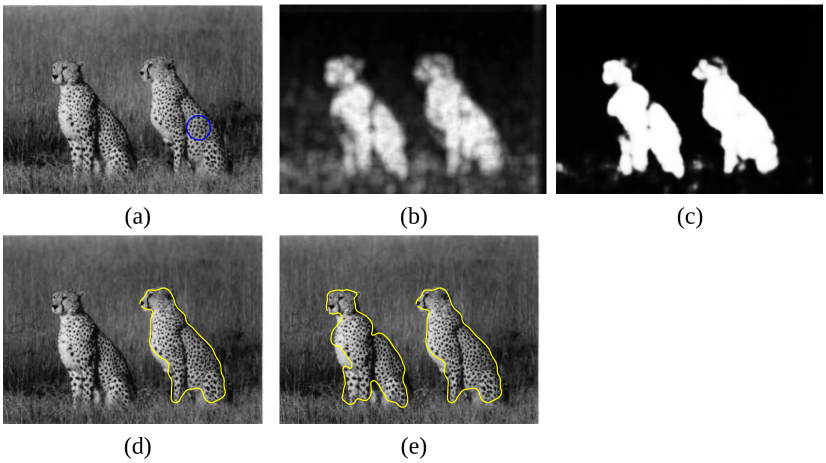

Fig. 7. (a) Original image with initial model. (b) Likelihood map based on Gabor response statistics. (c) Likelihood map after BP. (d) The converged model. (e) Both cheetah boundaries detected after initializing another model in the other high-likelihood area.

eral seed points inside the textured regions of interest, then a texture-consistency likelihood map is computed based on each model interior, the models evolve on their own dynamics, and those models with similar texture statistics are allowed to merge upon collision. The likelihood map for each model is re-computed after every 5 iterations of model evolution since the model interior statistics change as the model deforms. The balance factor k between the two energy terms in Eq. 15 is kept constant at $k = 200$, which is a value that is tested once and works well in all our experiments. The implementation of our algorithm is mostly in Matlab, but the most computationally expensive parts – the texon scale determination, likelihood map computation and BP based MRF – are implemented in C and linked to Matlab by CMex. The running time on a 2 GHz Pentium PC station for images of size 210×280 pixels is under 3 minutes, with two initial circular models of radius 10.

Fig. 7 is an experiment run on an image containing two cheetahs. The likelihood maps computed based on the initial model are shown, and the converged model finds

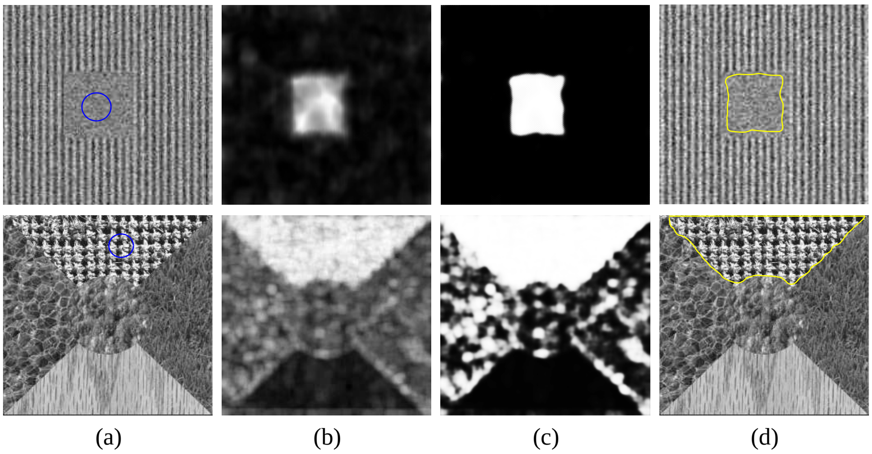

Fig. 8. (a) Original images. (b) Likelihood maps based on model-interior texture statistics. (c) Likelihood maps after BP. (d) The converged models at texture boundary.

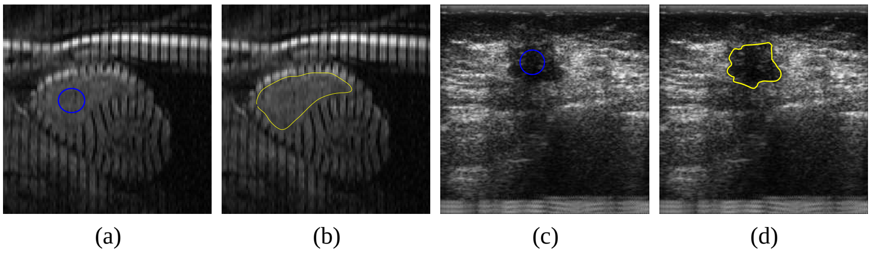

Fig. 9. (a) Original image and initial model for the tagged MR image. (b) segmentation result for tagged MR image. (c) original image and initial model for the ultrasound breast image. (d) segmented lesion in the breast image.

the boundary for one of the cheetahs. By initializing another model in another high-likelihood area, we are able to get the boundary for the other cheetah (Fig. 7(e)).

In Fig. 8, we demonstrate our algorithm using two synthetic images. The image on the top row has a small-scale homogeneous region in the center, and large-scale periodic line patterns in the background. The line pattern is generated using a sinusoidal signal. To test the robustness of the method to noise, we randomly added high level of gaussian noise to the entire image. The segmentation result shows that our method can deal with both small-scale and large-scale texture patterns, and has good differentiation power even in the presence of high noise levels. On the bottom row, we show the performance on a synthetic texture mosaic image. The image consists of five texture regions of similar intensity distribution, and we demonstrate the likelihood map and segmentation of one of the regions. We are able to segment the other four regions in the mosaic using the same method.

In Fig. 9, we show two examples of applying our model-based method to segment textured objects in medical images. On the left, we show segmentation of the right ventricle in a tagged MRI image of the heart. The result is shown in Fig. 9(b). And on the right is an ultrasound image of the breast. The goal is to segment the lesion on the breast. The final result is in Fig. 9(d). These results demonstrate the potential of our model-based method to deal with both large-scale tagging line patterns, and small-scale ultrasound speckle patterns.

5. Discussion and Conclusions

We have proposed a robust model-based segmentation method for finding boundaries of objects with complex textures of all scales and patterns. The novel ideas include an information-theoretic texon scale analysis algorithm and the online updating and learning of model-interior texture statistics to guide the model to achieve efficient, robust and accurate object segmentation.

Although we assume user-defined seed points to start the simple-shape initial models, our method can be directly applied to full-field image segmentation by starting multiple initial models on a regular lattice covering the image. The topology freedom of the models enables evolving models with similar statistics to merge, and finally the image is partitioned into regions of homogeneous textures.

References

1. S. M. Ali and S. D. Silvey. A general class of coefficients of divergence of one distribution from another. *J. Roy. Stat. Soc.*, 28:131–142, 1966.
2. J. F. Aujol, G. Aubert, and L. Blanc-Feraud. Wavelet-based level set evolution for classification of textured images. *IEEE Trans. on Image Processing*, 12(12):1634–1641, 2003.
3. V. Caselles, R. Kimmel, and G. Sapiro. Geodesic active contours. In *IEEE Int'l Conf. on Computer Vision*, pages 694–699, 1995.
4. H. Chernoff. Large-sample theory: Parametric case. *Ann. Math. Stat.*, 27:1–22, 1956.
5. D. Cohen. On active contour models and balloons. *CVGIP: Image Understanding*, 53:211–218, 1991.
6. T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. In *Proc. of European Conf. on Computer Vision*, volume 2, pages 484–498, 1998.
7. T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham. Active shape models - their training and application. *Computer Vision and Image Understanding*, 61(1):38–59, 1995.
8. J. Daugman. Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters. *Journal of the Optical Society of America A*, 2(7):1160–1169, 1985.
9. X. Huang, D. Metaxas, and T. Chen. Metamorphs: Deformable shape and texture models. In *IEEE Conf. on Computer Vision and Pattern Recognition*, volume 1, pages 496–503, 2004.
10. B. Julesz. Texons, the elements of texture perception, and their interactions. *Nature*, 290(5802):91–97, 1981.
11. T. Kailath. The divergence and bhattacharyya distance measures in signal selection. *IEEE Trans. on Comm. Tech.*, 15(1):52–60, 1967.
12. M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *Int'l Journal of Computer Vision*, 1:321–331, 1987.
13. J. Malik, S. Belongie, Leung T., and J. Shi. Contour and texture analysis for image segmentation. *Int'l Journal of Computer Vision*, 43(1):7–27, 2001.

14. R. Malladi, J. A. Sethian, and B. C. Vemuri. Shape modeling with front propagation: A level set approach. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 17(2):158–175, 1995.
15. B. Manjunath and Chellapa R. Unsupervised texture segmentation using markov random field models. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 13:478–482, 1991.
16. S. Osher and J. Sethian. Fronts propagating with curvature-dependent speed : Algorithms based on the Hamilton-Jacobi formulation. *Journal of Computational Physics*, 79:12–49, 1988.
17. N. Paragios and R. Deriche. Geodesic active regions for supervised texture segmentation. In *IEEE Int'l Conf. on Computer Vision*, pages 926–932, 1999.
18. N. Paragios, M. Rousson, and V. Ramesh. Matching Distance Functions: A Shape-to-Area Variational Approach for Global-to-Local Registration. In *European Conf. on Computer Vision*, pages II:775–790, 2002.
19. J. Pearl. *Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufman Publishers, 1988.
20. O. Pujol and P. Radeva. Texture segmentation by statistical deformable models. *Int'l Journal of Image and Graphics*, 4(3):433–452, 2004.
21. T. Randen and J. H. Husoy. Texture segmentation using filters with optimized energy separation. *IEEE Trans. on Image Processing*, 8(4):571–582, 1999.
22. C. Rother, V. Kolmogorov, and A. Blake. Grabcut: Interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics (SIGGRAPH)*, 23(3):309–314, 2004.
23. T. W. Sederberg and S. R. Parry. Free-form deformation of solid geometric models. In *Proceedings of the 13th Annual Conference on Computer Graphics*, pages 151–160, 1986.
24. J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.
25. L. H. Staib and J. S. Duncan. Boundary finding with parametrically deformable models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(11):1061–1075, 1992.
26. C. Xu and J. L. Prince. Generalized gradient vector flow external forces for active contours. *Signal Processing*, 71:131–139, 1998.
27. S. Zhu and A. Yuille. Region Competition: Unifying snakes, region growing, and Bayes/MDL for multi-band image segmentation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 18(9):884–900, 1996.
28. S. C. Zhu, C. E. Guo, Y. Z. Wang, and Z. J. Xu. What are textons? *Int'l Journal of Computer Vision*, 62(1):121–143, 2005.